03｜配好环境：选择一种最适合你的Go安装方法

2021-10-18 Tony Bai

《Tony Bai · Go语言第一课》 课程介绍 >

讲述：Tony Bai
时长 17:18 大小 15.86M

你好，我是 Tony Bai。

经过上一节的对 Go 设计哲学的探讨后，如果你依然决定继续 Go 语言编程学习之旅，那么欢迎你和我一起正式走进 Go 语言学习和实践的课堂。

编程不是“纸上谈兵”，它是一门实践的艺术。编程语言的学习离不开动手实践，因此学习任何一门编程语言的第一步都是要拥有一个这门编程语言的开发环境，这样我们才可以动手编码，理论与实践结合，不仅加速学习效率，还能取得更好的学习效果。

在这一讲中我们就先来学习下如何安装和配置 Go 语言开发环境。如果你的机器上还没有 Go，那么就请跟我一起选择一种适合你的 Go 安装方法吧。第一步，先来挑一个合适的 Go 版本。

选择 Go 版本

挑版本之前，我们先来看看 Go 语言的**版本发布策略**。

如今，Go 团队已经将版本发布节奏稳定在每年发布两次大版本上，一般是在二月份和八月份发布。Go 团队承诺对最新的两个 Go 稳定大版本提供支持，比如目前最新的大版本是 Go 1.17，那么 Go 团队就会为 Go 1.17 和 Go 1.16 版本提供支持。如果 Go 1.18 版本发布，那支持的版本将变成 Go 1.18 和 Go 1.17。支持的范围主要包括修复版本中存在的重大问题、文档变更以及安全问题更新等。

基于这样的版本发布策略，在你选择版本时可以参考这几种思路：

一般情况下，我建议你采用最新版本。因为 Go 团队发布的 Go 语言稳定版本的平均质量一直是很高的，少有影响使用的重大 bug。你也不用太担心新版本的支持问题，Google 的自有产品，比如 Google App Engine（以下简称为 GAE）支持都会很快，一般在 Go 新版本发布不久后，GAE 便会宣布支持最新版本的 Go。

你还可以根据不同实际项目需要或开源社区的情况使用不同的版本。

有的开源项目采纳了 Go 团队的建议，在 Go 最新版本发布不久就将当前项目的 Go 编译器版本升级到最新版，比如 Kubernetes 项目；而有的开源项目（比如：docker 项目）则比较谨慎，在 Go 团队发布 Go 1.17 版本之后，这些项目可能还在使用两个发布周期之前的版本，比如 Go 1.15。

但多数项目处于两者之间，也就是使用**次新版**，即最新版本之前的那个版本。比如，当前最新版本为 Go 1.17，那么这些项目会使用 Go 1.16 版本的最新补丁版本（Go 1.16.x），直到发布 Go 1.18 后，这些项目才会切换到 Go 1.17 的最新补丁版本（Go 1.17.x）。如果你不是那么“激进”，也可以采用这种版本选择策略。

因为我们这门课是 Go 语言学习的课，所以我这里建议你直接使用 Go 最新发布版，这样我们可以体验到 Go 的最新语言特性，应用到标准库的最新 API 以及 Go 工具链的最新功能。在这一节课里我们以 **Go 1.16.5 版本**为例讲述一下其安装、配置和使用方法。

选定 Go 版本后，接下来我们就来看看几种常见的 Go 安装方法。

安装 Go

Go 从 2009 年开源并演化到今天，它的安装方法其实都已经很成熟了，接下来呢，我们就逐一介绍在 Linux、macOS、Windows 这三大主流操作系统上安装 Go 的方法，我们假设这些操作系统都安装在 x86-64 的平台上，首先我们来看 Linux。

在 Linux 上安装 Go

Go 几乎支持 Linux 所有的主流发行版操作系统，常见的包括 Ubuntu、CentOS（Redhat 企业版 linux 的社区开源版）、Fedora、SUSE 等等，Go 在这些主流的 Linux 发行版操作系统上的安装方法都是一样的（当然某个发行版也可能会利用其软件安装管理器提供仅属于其自己的安装方法）。你可以参考下面这样的安装步骤。

首先，我们需要下载并解压 Go Linux 安装包：

复制代码

```
$wget -c https://golang.google.cn/dl/go1.16.5.linux-amd64.tar.gz
```

这里有个小提醒：虽然 Go 官方下载站点是 golang.org/dl，但我们可以用针对中国大陆的镜像站点 golang.google.cn/dl 来下载，在中国大陆地区使用大陆镜像站点可以大幅缩短下载时间。

第二步，将下载完毕的 Go 安装包解压到安装目录中：

复制代码

```
$tar -C /usr/local -xzf go1.16.5.linux-amd64.tar.gz
```

执行完上面解压缩命令后，我们将在 /usr/local 下面看到名为 go 的目录，这个目录就是 Go 的安装目录，也是 Go 官方推荐的 Go 安装目录。我们执行下面命令可以查看该安装目录下的组成：

复制代码

```
$ls -F /usr/local/go
AUTHORS          CONTRIBUTORS  PATENTS    SECURITY.md  api/  doc/         lib/
CONTRIBUTING.md  LICENSE       README.md  VERSION      bin/  favicon.ico  misc
```

不过呢，为了可以在任意路径下使用 go 命令，我们需要将 Go 二进制文件所在路径加入到用户环境变量 PATH 中（以用户使用 bash 为例），具体操作是将下面这行环境变量设置语句添加到 $HOME/.profile 文件的末尾：

```
1 export PATH=$PATH:/usr/local/go/bin
```

复制代码

然后执行下面命令使上述环境变量的设置立即生效：

```
1 $source ~/.profile
```

复制代码

最后，我们可以通过下面命令验证此次安装是否成功：

```
1 $go version
```

复制代码

如果这个命令输出了“go version go1.16.5 linux/amd64”，那么说明我们这次的 Go 安装是成功的。

在 Mac 上安装 Go

在 Mac 上我们可以在图形界面的引导下进行 Go 的安装。不过，我们先要下载适用于 Mac 的 Go 安装包：

```
1 $wget -c https://golang.google.cn/dl/go1.16.5.darwin-amd64.pkg
```

复制代码

安装包下载完毕后，我们可以双击安装包，打开标准的 Mac 软件安装界面，如下图所示：

按软件安装向导提示，一路点击“继续”，我们便可以完成 Go 在 Mac 上的安装。

和 Linux 一样，Mac 上的 Go 安装包默认也会将 Go 安装到 /usr/local/go 路径下面。因此，如果要在任意路径下使用 Go，我们也需将这个路径加入到用户的环境变量 PATH 中。具体操作方法与上面 Linux 中的步骤一样，也是将下面环境变量设置语句加入到 $HOME/.profile 中，然后执行 source 命令让它生效就可以了：

```
1 export PATH=$PATH:/usr/local/go/bin
```

复制代码

最后，我们同样可以通过 go version 命令验证一下这次安装是否成功。

当然了，在 Mac 上，我们也可以采用像 Linux 那样的通过命令行安装 Go 的方法，如果采用这种方法，我们就要下载下面的 Mac Go 安装包：

```
1 $wget -c https://golang.google.cn/dl/go1.16.5.darwin-amd64.tar.gz
```

复制代码

后续的步骤与 Linux 上安装 Go 几乎没有差别，你直接参考我上面讲的就好了。

在 Windows 上安装 Go

在 Windows 上，我们最好的安装方式就是采用图形界面引导下的 Go 安装方法。

我们打开Go 包的下载页面，在页面上找到 go 1.16.5 版本的 Windows msi 安装包（AMD64 架构下的）：go1.16.5.windows-amd64.msi，通过浏览器自带的下载工具它下载到本地任意目录下。

双击打开已下载的 go1.16.5.windows-amd64.msi 文件，我们就能看到下面这个安装引导界面：

和所有使用图形界面方式安装的 Windows 应用程序一样，我们只需一路点击“继续（next）”就可完成 Go 程序的安装了，安装程序默认会把 Go 安装在 **C:\Program Files\Go** 下面，当然你也可以自己定制你的安装目录。

除了会将 Go 安装到你的系统中之外，Go 安装程序还会自动为你设置好 Go 使用所需的环境变量，包括在用户环境变量中增加 GOPATH，它的值默认为 C:\Users[用户名]\go，在系统变量中也会为 Path 变量增加一个值：C:\Program Files\Go\bin，这样我们就可以在任意路径下使用 Go 了。

```
复制代码
1 安装完成后，我们可以打开Windows的“命令提示符”窗口（也就是cmd命令）来验证一下Go有没有安装成功
2
3 C:\Users\tonybai>go version
4 go version go1.16.5 windows/amd64
```

到这里，我们已经把 Go 语言在主流操作系统上的安装讲完了，但这里其实我讲的都是安装一个版本的 Go 的方法，有些时候我们会有安装多个 Go 版本的需求，这点我们接着往下看。

安装多个 Go 版本

一般来说，Go 初学者安装一个最新的 Go 版本就足够了，但随着 Go 编程的深入，我们通常会有使用多个 Go 版本的需求，比如一个版本用于学习或本地开发，另外一个版本用于生产构建等等。

安装多个 Go 版本其实也很简单，这里我给你介绍三种方法。

方法一：重新设置 PATH 环境变量

你只需要将不同版本的 Go 安装在不同路径下，然后将它们的 Go 二进制文件的所在路径加入到 PATH 环境变量中就可以了。

我们以 Linux 环境为例，在前面介绍 Go 标准安装方法的时候，我们已经将 go 1.16.5 版本安装到了 /usr/local/go 下面，也将 /usr/local/go/bin 这个路径加入到了 PATH 路径下了，当前状态我们在任意路径下敲入 go，执行的都是 go 1.16.5 版本对应的 Go 二进制文件。

那这个时候，如果我们想再安装一个 Go 1.15.13 版本要怎么办呢？

首先，你需要按照标准步骤将 go 1.15.13 安装到事先建好的 /usr/local/go1.15.13 路径下：

```
$mkdir /usr/local/go1.15.13
$wget -c https://golang.google.cn/dl/go1.15.13.linux-amd64.tar.gz
$tar -C /usr/local/go1.15.13 -xzf go1.15.13.linux-amd64.tar.gz
```

接下来，我们来设置 PATH 环境变量，将原先 $HOME/.profile 中的 PATH 变量的值由：

```
export PATH=$PATH:/usr/local/go/bin
```

改为：

```
export PATH=$PATH:/usr/local/go1.15.13/go/bin
```

这样通过执行 source 命令重新使 PATH 环境变量生效后，我们再执行 go version 命令，会得到下面这样的结果：

```
$go version
go version go1.15.13 linux/amd64
```

这样，我们已经安装好两个 Go 版本了。这之后，我们如果要在 Go 1.16.5 和 Go 1.15.13 两个版本之间切换，只需要重新设置 PATH 环境变量并生效即可。

不过，你可能依然会觉得通过重新设置 PATH 环境变量的方法有些麻烦。没关系，Go 官方也提供了一种在系统中安装多个 Go 版本的方法，下面我们就来看一下第二种方法。

方法二：go get 命令

这种方法有一个前提，那就是当前系统中已经通过标准方法安装过某个版本的 Go 了。

我们还以 Linux 环境为例，假设目前环境中已经存在了采用标准方法安装的 Go 1.16.5 版本，我们接下来想再安装一个 Go 1.15.13 版本。按照 Go 官方方法，我们可以这样来做：

首先，将 $HOME/go/bin$ 加入到 $PATH$ 环境变量中并生效，即便 HOME/go/bin 这个目录当前不存在也没关系：

复制代码

```
export PATH=$PATH:/usr/local/go/bin:~/go/bin
```

然后，我们要执行下面这个命令安装 Go 1.15.13 版本的下载器：

复制代码

```
$go get golang.org/dl/go1.15.13
```

这个命令会将名为 go1.15.13 的可执行文件安装到 $HOME/go/bin 这个目录下，它是 Go 1.15.13 版本的专用下载器，下面我们再来执行 Go 1.15.13 的下载安装命令：

复制代码

```
$go1.15.13 download

Downloaded   0.0% (    16384 / 121120420 bytes) ...
Downloaded   1.8% (  2129904 / 121120420 bytes) ...
Downloaded  84.9% (102792432 / 121120420 bytes) ...
Downloaded 100.0% (121120420 / 121120420 bytes)
Unpacking /root/sdk/go1.15.13/go1.15.13.linux-amd64.tar.gz ...
Success. You may now run 'go1.15.13'
```

现在，我们看到这个命令下载了 go1.15.13.linux-amd64.tar.gz 安装包，也将它安装到 $HOME/sdk/go1.15.13 下面了。下载安装结束后，我们就可以利用**带有版本号的 go 命令**来使用特定版本的 Go 了：

复制代码

```
$go1.15.13 version
go version go1.15.13 linux/amd64
```

同样的，我们也可以通过下面这个命令查看特定 Go 版本的安装位置：

复制代码

```
$go1.15.13 env GOROOT
/root/sdk/go1.15.13
```

方法三：go get 命令安装非稳定版本

其实，除了 Go 团队正式发布的稳定版本 (stable version)，像前面安装的 Go 1.16.5 或 Go 1.15.13，我们还可以通过 **go get** 的方法安装 Go 团队正在开发的非稳定版本 (Unstable Version)，包括每个稳定版发布前的 beta 版本或当前最新的 tip 版本，这些非稳定版本可以让 Go 开发人员提前体验到即将加入到稳定版本中的新特性。

但是，通过 go get 安装不同 Go 版本的方法在中国大陆地区会因网络问题而失败。如果你已经克服了网络问题，那安装非稳定版本的步骤其实和上面的步骤一样。现在，我们以 Go 1.17beta1 和 Go Tip 版本为例，带你体验一下它们的安装步骤和验证方法。

首先我们来看 Go 1.17beta1：

复制代码

```
$go get golang.org/dl/go1.17beta1
$go1.17beta1 download
Downloaded   0.0% (     3272 / 134470397 bytes) ...
Downloaded  21.4% ( 28819248 / 134470397 bytes) ...
Downloaded  58.1% ( 78069168 / 134470397 bytes) ...
Downloaded 100.0% (134470397 / 134470397 bytes)
Unpacking /root/sdk/go1.17beta1/go1.17beta1.linux-amd64.tar.gz ...
Success. You may now run 'go1.17beta1'
$go1.17beta1 version
go version go1.17beta1 linux/amd64
```

接着来看 Go Tip 版本：

复制代码

```
$go get golang.org/dl/gotip
$gotip download
```

go get 为我们安装 tip 版本提供了极大方便，要知道在以前，如果我们要安装 tip 版本，需要手工下载 Go 源码并自行编译。但你要注意的是：**不是每次 gotip 安装都会成功**，因为这毕竟是正在积极开发的版本，一次代码的提交就可能会导致 gotip 版本构建失败！

安装好 Go 之后，我们就该讲讲怎么配置了。

配置 Go

其实 Go 在安装后是开箱即用的，这也意味着我们在使用 Go 之前无需做任何配置。但为了更好地了解和学习 Go，我们还是要认识一些 Go 自带的常用配置项。Go 的配置项是以环境变量的形式存在的，我们可以通过下面这个命令查看 Go 的这些配置项：

复制代码

```
$go env
```

这里我也给你总结了一些常用配置项：

名称	作用	值
GOARCH	用于指示Go编译器生成代码所针对的平台CPU架构	主要值是AMD64、Arm等，默认值是本机的CPU架构
GOOS	用于指示Go编译器生成代码所针对的操作系统	主要值是Linux、Darwin、Windows等，默认值是本机的操作系统
GO111MODULE	它的值决定了当前使用的构建模式是传统的GOPATH模式还是新引入的Go Module模式	在Go 1.16版本Go Module构建模式默认开启，该变量值默认为**on**
GOCACHE	用于指示存储构建结果缓存的路径，这些缓存可能会被后续的构建所使用	在不同的操作系统上，GOCACHE有不同的默认值。以Linux为例，我们使用go env GOCACHE查看其值为：$HOME/.cache/go-build
GOMODCACHE	用于指示存放Go Module的路径	在不同的操作系统上，GOMODCACHE有不同的默认值。以Linux为例，我们使用go env GOMODCACHE查看其值为：$HOME/go/pkg/mod
GOPROXY	用来配置Go Module proxy服务	其默认值为“**https://proxy.golang.org,direct**”。中国大陆地区，一般设置为大陆地区提供的module proxy服务以加速Go Module的获取速度。值可设置为“**https://goproxy.cn,direct**”
GOPATH	在传统的GOPATH构建模式下，用于指示Go包搜索路径的环境变量，在Go module机制启用之前是Go核心配置项。Go 1.8版本之前需要手工配置，Go 1.8版本引入了默认的GOPATH（$HOME/go）。在Go Module模式正式上位后，很可能会被移除	
GOROOT	指示Go安装路径。Go 1.10版本引入了默认GOROOT，开发者无需显式设置，Go程序会自动根据自己所在路径推导出GOROOT的路径	

如果你还要了解更多关于 Go 配置项的说明，你可以通过 go help environment 命令查看。

小结

好了，到这里我们的 Go 安装配置方法就讲解完毕了，选好你要使用的 Go 安装方法了吗？

在这一节课中我们首先讲解了三种 Go 版本的选择策略：

第一种，也是我们推荐的一种，那就是使用 Go 最新的版本，这样你可以体验到 Go 的最新语言特性，应用到标准库的最新 API 以及 Go 工具链的最新功能，并且很多老版本中的 bug 在最新版本中都会得到及时修复；

如果你还是对最新版本的稳定性有一丝担忧，你也可以选择使用次新版；

最后，如果你要考虑现存生产项目或开源项目，那你按照需要选择，与项目策略保持一致就好了。

确定完 Go 版本后，我们就可以来安装这个 Go 版本了。这一节课我们也详细介绍了在三个主流操作系统上安装 Go 稳定版本的方法。

对于使用 Windows 或 macOS 操作系统的开发者，使用基于图形界面的安装方式显然是最方便、最简洁的；对于使用 Linux 操作系统的开发者，使用自解压的安装包，或者是通过操作系统自带安装工具来进行 Go 安装比较普遍。

如果你是要在本地开发环境安装多个 Go 版本，或者是要抢先体验新版 Go，我们还讲解了两种在本地安装多个 Go 版本的方法。这里再强调一下，通过 go get 方式安装最新的 Go tip 版本存在失败的可能性哦！

最后，我们讲解了 Go 的一些常用配置项的功用，对于中国地区的 Go 开发者而言，你在真正使用 Go 构建应用之前，唯一要做的就是配置 GOPROXY 这个 Go 环境变量。

有了 Go 开发环境，我们就有了编写和构建 Go 代码的基础，在下一讲中我们就将开始学习如何编写 Go 代码。

思考题

今天的课后思考题，我想请你在安装好的 Go 开发环境中，使用 go help 命令查看和总结一下 Go 命令的使用方法。

感谢你和我一起学习，也欢迎你把这节课分享给更多对 Go 语言学习感兴趣的朋友。我是 Tony Bai，我们下节课见。

分享给需要的人，Ta订阅后你可得 20 元现金奖励

生成海报并分享

赞 1 提建议

© 版权归极客邦科技所有，未经许可不得传播售卖。 页面已增加防盗追踪，如有侵权极客邦将依法追究其法律责任。

上一篇 02 | 拒绝“Hello and Bye”：Go语言的设计哲学是怎么一回事？

精选留言 (7)

写留言

陈彬

2021-10-18

Tony老师
$go get golang.org/dl/go1.15.13

这个是不是已经不能用了，改成了：
$go install golang.org/dl/go1.15.13@latest

展开

宝仔

2021-10-18

老师多版本管理工具gvm怎么样？可以用于生产环境吗

陈东

2021-10-18

1、按要求操作，go help
2、总结：
go build主要用于编译代码。
go clean移除当前源码包和关联源码包。
3、安装好环境后，不设置PATH，再安装Goland jetbrains，能不能自动关联已安装好的p...

展开

一步

2021-10-18

go 1.17 执行 go get go 已经安装不了了
go get: installing executables with 'go get' in module mode is deprecated.
Use 'go install pkg@version' instead.
For more information, see https://golang.org/doc/go-get-install-deprecation
or run 'go help get' or 'go help install'.

展开

Geek_95004c
2021-10-18

没有听，直接看完的，感觉说不出来，等后续

展开

剑龙
2021-10-18

老师，go的环境产量设置应该和系统环境变量是不同的吧，包括设置方式和查看方式？

Hurt
2021-10-18

老师 需要配置GoPath吗

展开

1